(19) Europäisches Patentamt
European Pat nt Offic
Office europ´en des brevets

(11) EP 0 908 833 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
14.04.1999 Bulletin 1999/15

(51) Int Cl.^{6}: G06F 17/30

(21) Application number: 98307994.8

(22) Date of filing: 30.09.1998

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: 11.10.1997 GB 9721502

(71) Applicant: INTERNATIONAL COMPUTERS LIMITED
Putney, London, SW15 1SW (GB)

(72) Inventor: Tinkler, Simon Paul
Kingston Upon Thames, Surrey KT1 2JE (GB)

(74) Representative: Guyatt, Derek Charles et al
Intellectual Property Department
International Computers Limited
Cavendish Road
Stevenage, Herts, SG1 2DY (GB)

(54) Computer system for supporting a website

(57) A computer system for supporting a website, comprises a navigation object table defining a number of navigation objects, representing a tree-like navigational structure. and an object handler which uses the table dynamically to construct menus allowing a user to navigate through the navigational structure. Whenever the object handler is called, it is passed a tag identifying an object. It then searches the navigation object table to find objects whose parent object in the tree structure is the object identified by this tag, using these objects to construct the menu. The navigation objects include menu objects specifying subsidiary menus in the tree structure, and selection of one of the menu objects causes a recursive call to the object handler, passing it th tag of the menu object. The system also includes an access control table indicating which users may access which navigation objects, and the object handler uses this access control table to tailor the menus for particular users.

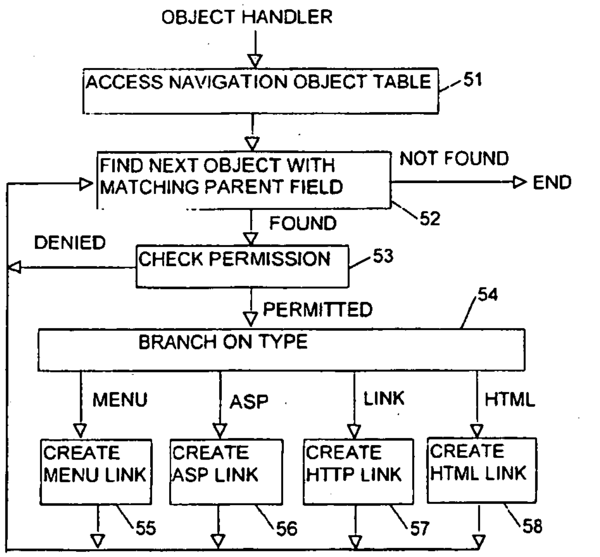

FIG. 5

Printed by Jouve, 75001 PARIS (FR)

Description

Background to the Invention

[0001] This invention relates to a computer system for supporting a website on the Internet or on an intranet.

[0002] Normally, a website consists of a number of pages, linked together by means of hyperlinks, allowing a user to navigate around from one page to another. However, a problem with this is that, when pages are created or updated, it is very difficult to ensure that all the necessary links are correctly updated in a consistent manner. For example, if a file is moved to a different location, all links to the file must be changed so that they now point to the new location. The object of the present invention is to provide a way of overcoming or alleviating this problem.

[0003] Websites are frequently used to provide information and services to employees of an organisation, to enable them to carry out their jobs more effectively. However, if a website is large, and contains a large amount of information not relevant to a particular employee's job, it can be difficult for the employee to locate the information he or she requires. An object of a preferred form of the present invention is to provide a way of overcoming or alleviating this problem.

Summary of the Invention

[0004] According to the invention there is provided a computer system for supporting a website, characterised by:

(a) a navigation object table defining a plurality of navigation objects representing a navigational structure for navigating around the website; and
(b) an object handler for using said navigation table dynamically to construct menus allowing a user to navigate through said navigational structure.

[0005] As will be shown, the invention enables the structure of the website to be modified. simply by altering the contents of the navigation object table, without having to alter links in the website pages.

[0006] In a preferred form of the invention, the system also includes an access control table indicating which users may access which navigation objects, and wherein the object handler uses this access control table to tailor the menus for particular users.

Brief Description of the Drawings

[0007] Figure 1 is a block diagram of an intranet, including a computer system in accordance with the present invention.

[0008] Figure 2 is a schematic diagram showing the structure of a set of object tables used in the computer system.

[0009] Figure 3 is a schematic diagram showing the tree structure of navigational objects defined by the object tables.

[0010] Figur 4 is a diagram illustrating the format of a home page screen display produced by the system.

[0011] Figure 5 is a flow chart showing the operation of an object handler forming part of the website.

[0012] Figure 6 is a diagram showing a screen display containing a menu dynamically constructed by the object handler.

Description of an Embodiment of the Invention

[0013] One embodiment of the invention will now be described by way of example with reference to the accompanying drawings.

Overview

[0014] Figure 1 shows an intranet system, comprising a server computer 101, and a number of client computers 102, interconnected by means of a network 103.

[0015] The server computer 101 runs a server manager 104, and includes a file store 105 and a database 106. In this example, the server computer runs under the Microsoft Windows NT Version 4 operating system, the server manager is Microsoft Internet Information Server (IIS) version 3.00, while the database is provided by Microsoft SQL Server version 6.5.

[0016] The server computer maintains a website, comprising a large number of pages, held in the file store 105. These pages include HTML (Hypertext Markup Language) files 107, and ASP (Active Server Page) files 108. An ASP file is an example of a script file, running under an interpreter, which can perform various functions when called. For example, an ASP file may contain embedded SQL queries, for accessing the database.

[0017] Each of the client computers 102 includes a web browser 115 which, in this example, is Microsoft Internet Explorer.

[0018] The navigational structure of the website is represented by a number of navigation objects, arranged in a tree structure. These objects are defined by a set of object tables, held in the database 106. Thes object tables comprise: Navigation Object Table 109, ASP Object Table 110, Link Object Table 111, and HTML Object Table 112. Figure 2 shows a simplified example of these tables, while Figure 3 shows the corresponding tree structure.

Navigation object table

[0019] The navigation object table 109 holds an entry for each navigation object. Each entry contains the following fields:

TAG An identifying tag value. The TAG field of each object is formed by adding three digits to the tag value of its parent object in the tree structure. For example, the object TEC001 is the parent of objects TEC001001 and TEC001002.

NAME Text for display on a link.

TYPE In the present example, the following object types are defined, although it should be appreciated that other types may also be defined:

MENU Menu.
ASP ASP (Active Server Page).
LINK A link to another website.
HTML HTML (Hypertext Markup Language).

ACTIVE"True" if the object is currently active.

PARENTThe tag of the parent object in the tree structure. (This field is not strictly necessary, since it can be deduced from the TAG value).

ASP Object Table

[0020] The ASP Object Table 110 holds an entry for each ASP file that can be accessed via the navigational structure. Each entry contains the following fields:

TAG A tag value which relates the entry to an ASP object in the Navigation Object Table.

FILENAME The file name of the ASP file.

Link Object Table

[0021] The Link Object Table 111 holds an entry for each remote website that can be accessed via the navigational structure. Each entry contains the following fields:

TAG A tag value which relates the entry to a LINK object in the Navigation Object Table.

ADDRESS The full HTTP address of the remote website.

HTML Object Table

[0022] The HTML Object Table 112 holds an entry for each HTML file that can be accessed via the navigational structure. Each entry contains the following fields:

PARENT A tag value which relates the entry to an HTML object in the Navigation Object Table.

FILENAME The file name of the HTML file.

NAME Text for display on a link.

[0023] It should be noted that several entries in the HTML Object Table may share the same PARENT value. This allows a group of HTML files to be represented by a single entry in the Navigation Object Table.

Access control tables

[0024] The database also holds two access control tables: NavNTGroup table 113, and NTGroup Table 114. These tables are used to control the way in which individual users can access given navigation objects.

[0025] The NavNTGroup table 113 associates navigation objects with groups which are permitted to access those objects. This table contains the following fields;

TAG A tag value, identifying a navigation object.

NTGROUP An NT group name.

[0026] A given tag value may be associated with more than one NT group, allowing more than one group of users to access a particular navigation object.

[0027] By default, if no entry appears for a particular tag value, the corresponding navigation object can be accessed by any group.

[0028] The NTGroup table 114 associates users with specific groups. This table contains the following fields:

USERNAME A username.

NTGROUP An NT group name.

[0029] A given username may be associated with several different groups.

User attachment to the website

[0030] A user can attach to the website by entering the web address of the website into the browser 115. The browser will then contact the server manager 104, to inform it that the user is attaching to the website.

[0031] When the user attaches to the website, a file DEFAULT.HTM is automatically loaded by the server manager. This is a static HTML file which sets up two frames on the user's screen. As shown in Figure 4, these two frames comprise a navigation bar 41 and a main window 42. The navigation bar 41 consists of an area running horizontally across the top of the screen, while the main window consists of the remainder of the screen. DEFAULT.HTM simply composes these two frames; the content of the frames is built by calling two further files NAVBAR.HTM and MAINHARD.ASP.

[0032] NAVBAR.HTM is a static HTML file. It paints the navigation bar 41, setting up a number of navigation buttons 43 in this bar.

[0033] MAINHARD.ASP paints the main window 42 for the home page of the website. As shown in Figure 4, this includes a number of zone buttons 44, representing the top-level zones of the site. The main window may also include a graphic 45, such as a company logo.

[0034] The home page also includes a button which activates a script which performs a user log-in procedure with the server, allowing the server to obtain the user's username.

Zone buttons

[0035] Each of the zone buttons has a tag associated with it. For example, the following zone buttons may be provided, with tags as shown:

Zone	Tag
Knowledge Modules	KNO
Operational support	OPS
Technical Support	TEC
Management Information	MAN
Personnel and Development	PDI
Sales and Marketing	SAM
Policy	POL
Webmaster	WEB

[0036] The zone buttons 44 form the highest level of the navigational tree, and are the parents of the top-level objects in the Navigation Object table. For example, it can be seen that in Figure 2, the "Technical support" zone button (TEC) is the parent of the first three items in the Navigation Object table.

[0037] Whenever one of the zone buttons is selected by the user, MAINHARD.ASP calls an object handler, HANDLE_OBJECT.ASP, passing it the tag associated with that button.

Object handler

[0038] Figure 5 shows the operation of the object handler when it is called from MAINHARD.ASP, or recursively called by itself.

[0039] (Step 51) The object handler accesses the navigation object table in the database.

[0040] (Step 52) The object handler searches for the next navigation object in the navigation object table whose PARENT field matches the tag value passed to the object handler when it was called. If no matching object can be found, the object handler terminates.

[0041] (Step 53) If a matching object is found, the object handler then consults the access control tables NavNTGroup and NTGroup, to check whether the current user is a member of a group which is permitted to access this navigation object. If access is not permitted, the object handler returns to step 52 above.

[0042] (Step 54) Assuming that access is permitted, the object handler then branches according to the TYPE field of the navigation object.

[0043] (Step 55) If the navigation object is of the MENU type, the object handler creates a link in the main window 42, using the NAME field of the navigation object as the label for the link, and associating th TAG value of the navigation object with this link. If this link is selected by the user, the object handler will recursively call itself, passing itself the associated TAG value.

[0044] (Step 56) If the navigation object is of the ASP type, the object handler accesses the ASP Object Table, to find the entry whose TAG matches that of the navigation object, and reads the FILENAME field of this entry. The object handler then creates a link to this FILENAME in the main window 42 , using the NAME field of the navigation object as the label for the link.

[0045] (Step 57) If the navigation object is of the LINK type, the object handler accesses the Link Object Table, to find the entry whose TAG matches that of the navigation object, and reads the ADDRESS field of this entry. The object handler then creates a link to this ADDRESS in the main window 42, using the NAME field of the navigation object as the label for the link.

[0046] (Step 58) If the navigation object is of the HTML type, the object handler accesses the HTML Object Table, and searches for each entry whose PARENT field matches the TAG value of the navigation object. For each matching entry in the HTML Object Table, the object handler reads the FILENAME field, and then creates a link to this FILENAME in the main window 42, using the NAME field from the HTML Object Table entry as the label for the link.

[0047] In each of the steps 55-58, if the navigation object has its ACTIVE field equal to "true", the link is displayed as an active link, conventionally in blue. On the other hand, if the navigation object has its ACTIVE field equal to "false", the link is displayed as an inactive link, conventionally in grey.

[0048] After performing any of steps 55-58, the object handler returns to step 52 above. The loop (steps 52-58) is then repeated until all matching objects have been processed.

[0049] It can be seen that, when called, the object handler dynamically builds a menu, consisting of a set of links, using the information contained in the object tables in the database. The links may provide access to particular files or websites, or may lead to a lower level menu.

[0050] For example, if the user selects the "Technical Support" zone button from the home page, the object handler will search the Navigation Object Table to find all entries whose PARENT field matches the tag valu TEC of this button. In the example shown in Figure 2, the first entry matches, and so the object handler will create a link with the text "useful websites", and the tag value TEC001. Similarly, the second entry matches, and so the object handler will also create a link to the file skills.asp with the text "skills database". The third entry also matches, and in this case the object handler will create two more links, one for each of the HTML files whose PARENT field matches the tag TEC003. Thus, the object handler builds a menu with four items, as shown in Figure 6.

[0051] The user can then select any of the links in the menu. For example, if the user selects the "useful websites" link, the object handler will call itself recursively, passing itself the tag value TEC001. In response to this call, the object handler builds a further menu containing a set of links to remote websites. Alternatively, if the "skills database" link is selected, the skills.asp file is called, or if any of the HTML links is selected, the corresponding HTML file is called.

Discussion

[0052] It has been shown that the object handler, when called, dynamically builds a menu, using the information contained in the object tables in the database. The structure of the website can therefore easily be modified, simply by altering the contents of the object tables in the database. It is not necessary to alter links within the files themselves. This greatly eases the task of updating the website.

[0053] Each menu can be individually tailored to the user, by means of the access control tables, so as to contain only those items that are relevant to the particular user's job. However the user still has the ability gain access to resources in the conventional manner, by using the GO function of the browser, bypassing the navigational structure provided by the object tables. This is useful in cases where the user normally works within a fixed set of information, but may occasionally require to access other resources.

[0054] As described above, the server obtains the user's username through a log-in procedure. If this procedure is not performed, a default username is assumed. This default username allows access only to a default subset of resources.

Some possible modifications

[0055] It will be appreciated that many modifications may be made to the system described above without departing from the scope of the present invention. For example, other types of navigation object may be defined. Also, although the above example was described in the context of an intranet, it will be appreciated that the invention could also be employed for Internet access.

[0056] Although the embodiment of the invention described above was implemented using Microsoft technology, it will be appreciated that this is not essential to the invention, and other embodiments of the invention may be implemented using different technology.

Claims

1. A computer system for supporting a website, characterised by:

   (a) a navigation object table (109) defining a plurality of navigation objects representing a navigational structure for navigating around the website; and
   (b) an object handler (51-58) for using said navigation object table dynamically to construct menus allowing a user to navigate through said navigational structure.

2. A computer system according to Claim 1 wherein said navigation objects are arranged in a tree structure, and wherein said object handler includes:

   (a) means (52) for searching the navigation object table to find objects having a specified parent object in the tree structure; and
   (b) means (54-58) for using the objects so found to construct a menu.

3. A computer system according to Claim 2 wherein said navigation objects include menu objects, and wherein said object handler includes means responsive to selection of one of said menu objects for recursively calling said object handler to construct a subsidiary menu.

4. A computer system according to any preceding claim further including:

   (a) an access control table (113,114) indicating which users may access which navigation objects; and
   (b) means (53) for using said access control table to tailor the menus for particular users.

5. A computer system according to any preceding claim wherein said navigation objects include HTML objects specifying links to HTML files, and script objects specifying links to script files.

6. A computer system according to any preceding claim wherein said navigation objects include link objects specifying links to remote websites.

7. A method of operating a computer system to support a website, the method comprising:

   (a) creating a navigation object table defining a plurality of navigation objects representing a navigational structure for navigating around the website; and
   (b) using said navigation object table dynamically to construct menus allowing a user to navigate through the navigational structure.

8. A method according to Claim 7 including the steps; (a) arranging the navigation objects in a tree structure; (b) searching the navigation object table to find

objects having a specified parent object in the tree structure; and (c) using the objects so found to construct a menu.

9. A method according to Claim 8 wherein said navigation objects include menu objects, and including the step of responding to selection of one of said menu objects by recursively performing the steps specified in Claim 9 to construct a subsidiary menu.

10. A method according to any one of Claims 7 to 9 including the steps:

(a) providing an access control table indicating which users may access which navigation objects; and
(b) using said access control table to tailor the menus for particular users.

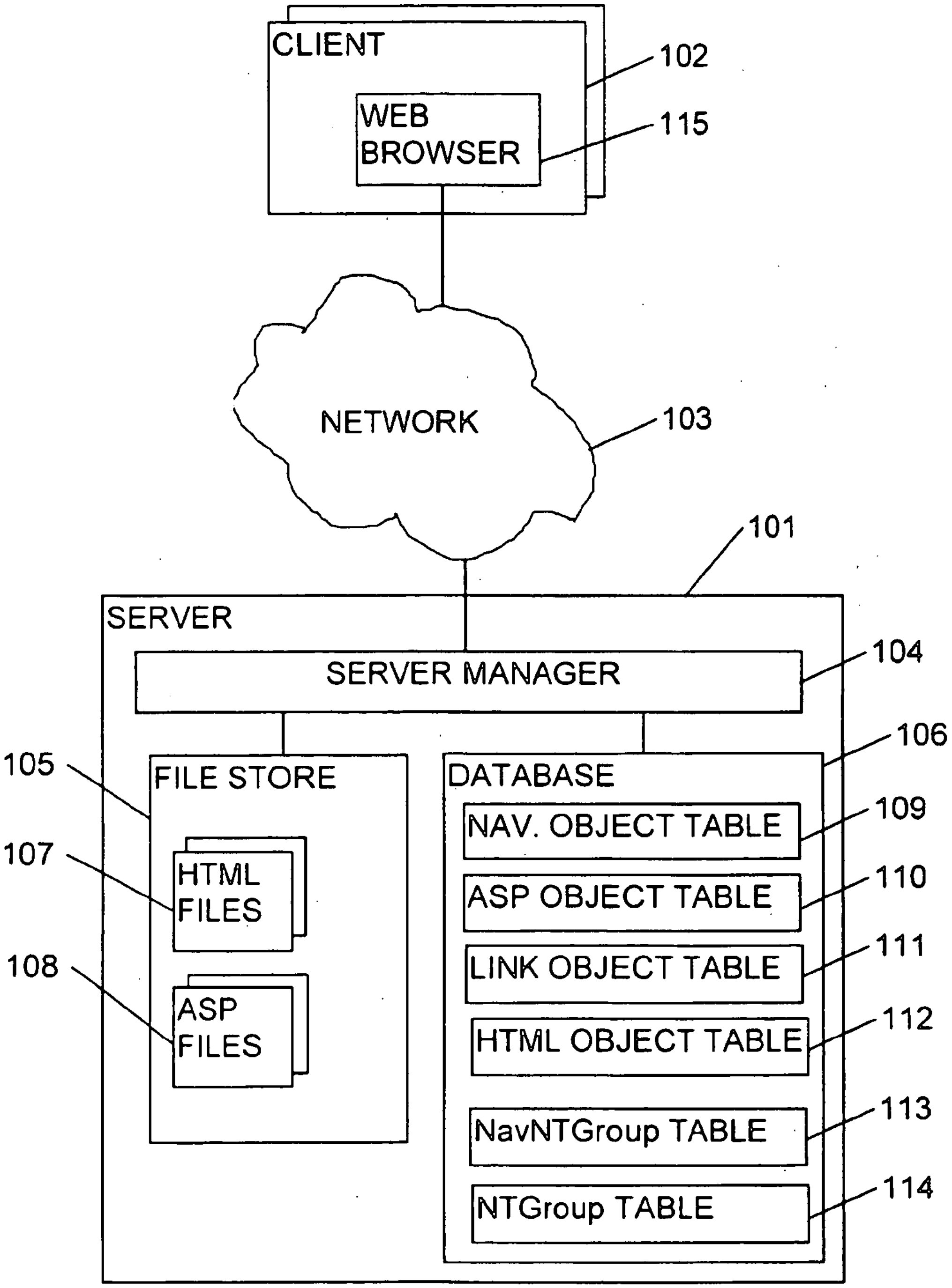
CLIENT
WEB BROWSER
102
115
NETWORK
103
101
SERVER
SERVER MANAGER
104
105
FILE STORE
HTML FILES
107
ASP FILES
108
DATABASE
106
NAV. OBJECT TABLE
109
ASP OBJECT TABLE
110
LINK OBJECT TABLE
111
HTML OBJECT TABLE
112
NavNTGroup TABLE
113
NTGroup TABLE
114

FIG.1

NAVIGATION OBJECT TABLE

TAG	NAME	TYPE	ACTIVE	PARENT
TEC001	useful websites	MENU	T	TEC
TEC002	skills database	ASP	T	TEC
TEC003	information	HTML	T	TEC
TEC001001	Microsoft	LINK	T	TEC001
TEC001002	Oracle	LINK	T	TEC001

ASP OBJECT TABLE

TAG	FILENAME
TEC002	skills.asp

LINK OBJECT TABLE

TAG	ADDRESS
TEC001001	http://www.microsoft.com
TEC001002	htttp://www.oracle.com

HTML OBJECT TABLE

PARENT	FILENAME	NAME
TEC003	cap.htm	Capabilities
TEC003	prod.htm	Products

FIG. 2

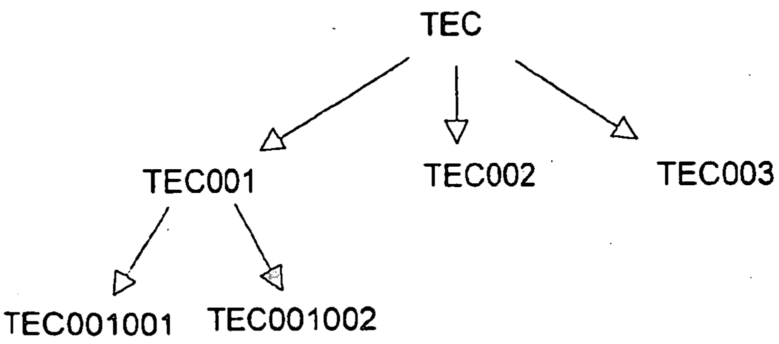
TEC
TEC001
TEC002
TEC003
TEC001001
TEC001002

FIG. 3

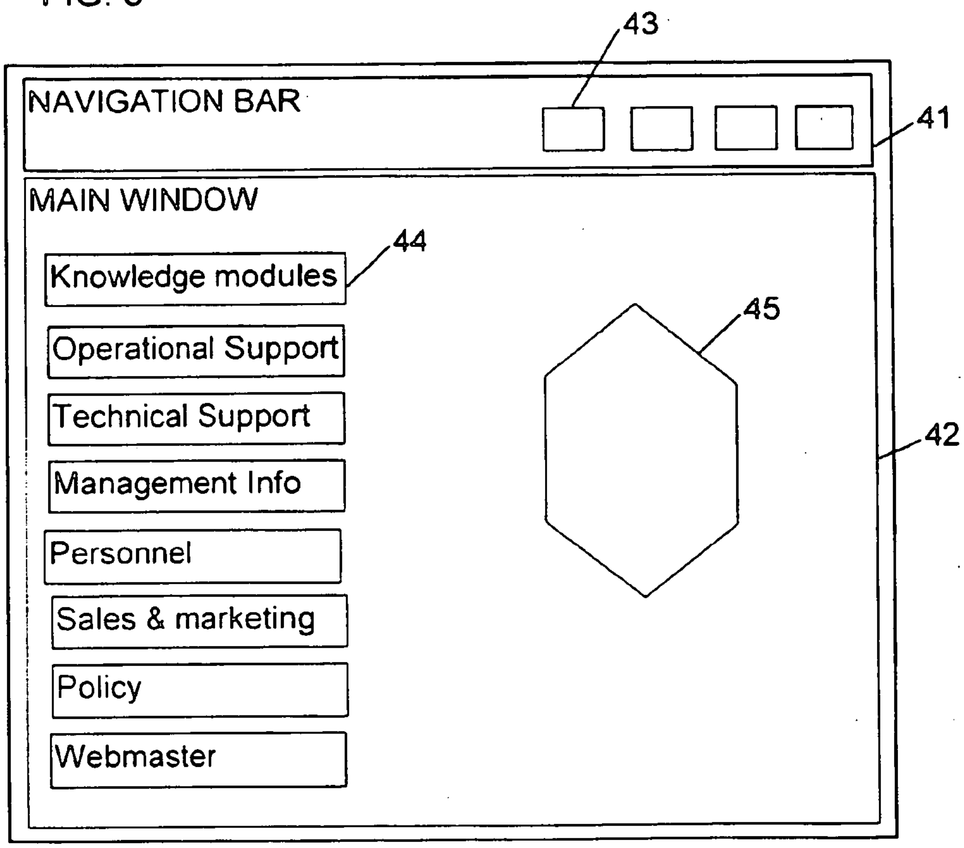
43
NAVIGATION BAR
41
MAIN WINDOW
44
Knowledge modules
Operational Support
Technical Support
Management Info
Personnel
Sales & marketing
Policy
Webmaster
45
42

FIG. 4

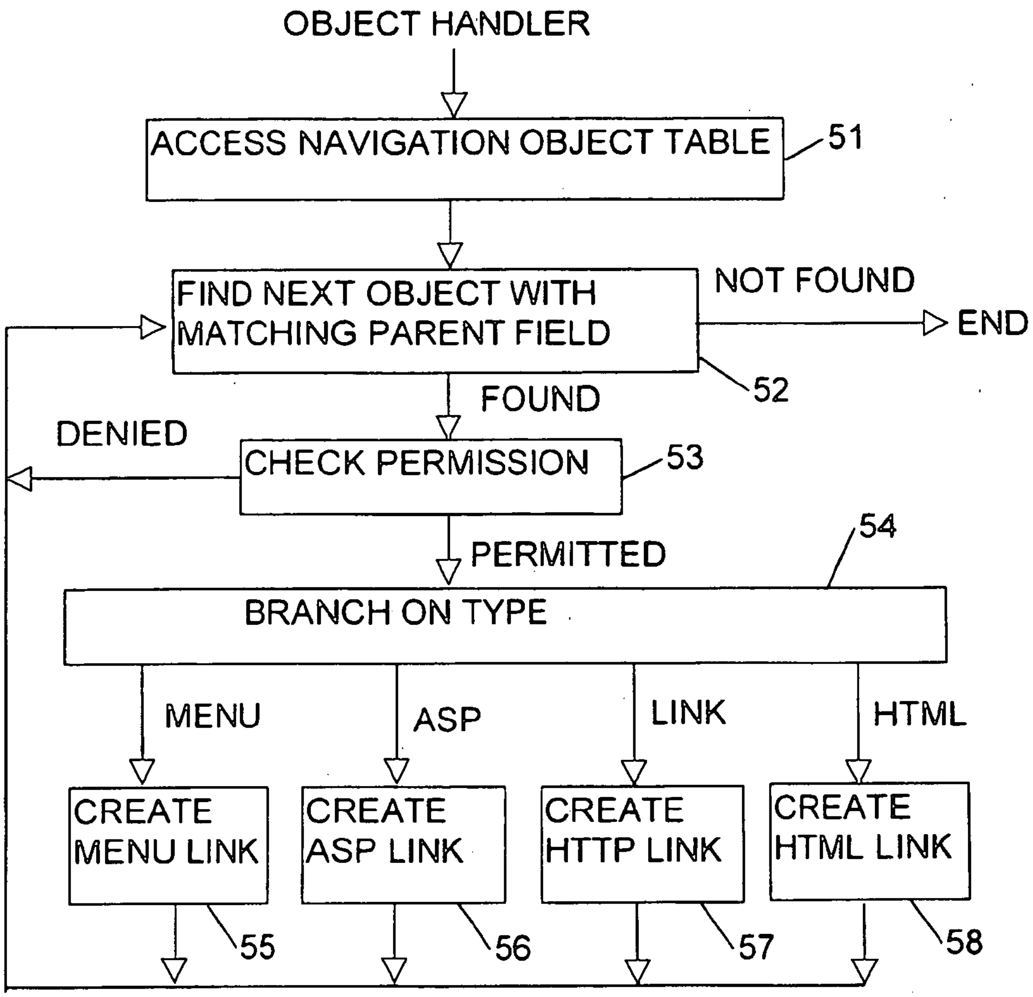
OBJECT HANDLER
ACCESS NAVIGATION OBJECT TABLE
51
FIND NEXT OBJECT WITH MATCHING PARENT FIELD
NOT FOUND
END
52
FOUND
DENIED
CHECK PERMISSION
53
PERMITTED
54
BRANCH ON TYPE
MENU
ASP
LINK
HTML
CREATE MENU LINK
CREATE ASP LINK
CREATE HTTP LINK
CREATE HTML LINK
55
56
57
58

FIG. 5

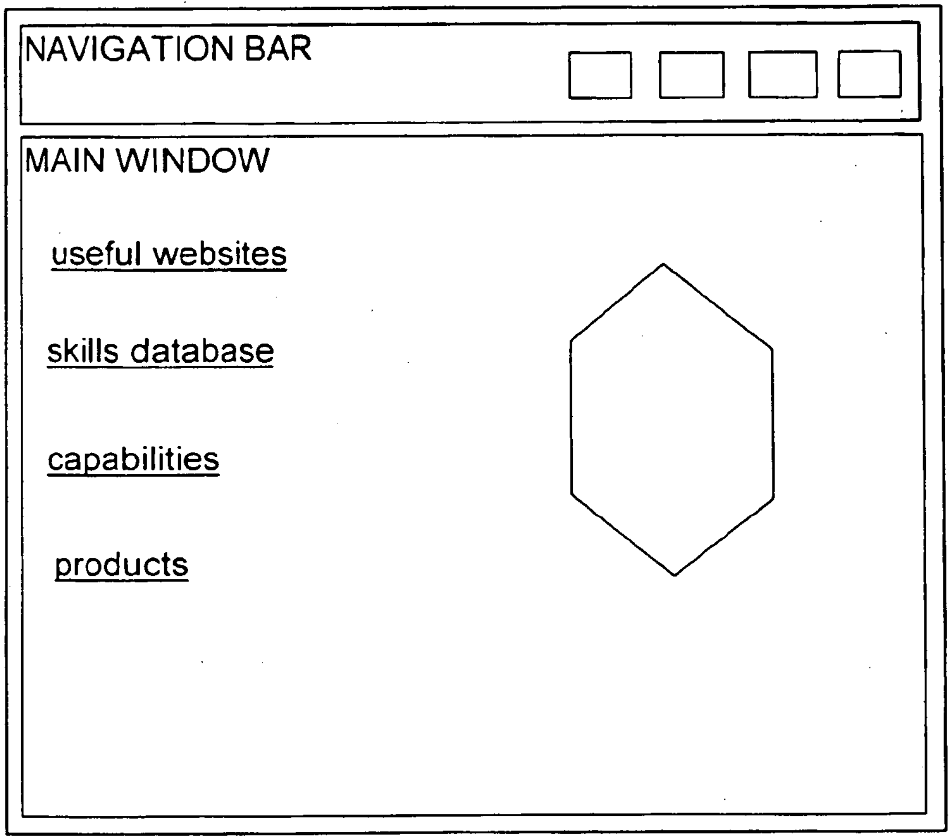
NAVIGATION BAR
MAIN WINDOW
useful websites
skills database
capabilities
products

FIG. 6